TITLE:

# Policy-Aware Parallel Execution of Composite Services( Abstract_要旨 )

AUTHOR(S):

Mai, Xuan Trang

CITATION:

Mai, Xuan Trang. Policy-Aware Parallel Execution of Composite Services. 京都大学, 2016, 博士(情報学)

ISSUE DATE:

2016-03-23

URL:

https://doi.org/10.14989/doctor.k19855

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

（ 続紙 1 ）

| 京都大学 | 博士（情報学） | 氏名 | MAI XUAN TRANG |
|---|---|---|---|
| 論文題目 | Policy-Aware Parallel Execution of Composite Services<br>（複合サービスのポリシーアウェアな並列実行） | | |

（論文内容の要旨）

The goal of this thesis is to enhance processing efficiency of parallel execution for web services. To this end, it proposes a new approach to model a service provider's policy of parallel execution and to allow service users to execute web services with an optimal degree of parallelism. This thesis consists of six chapters.

Chapter 1 outlines the thesis, including the research objective, issues and approaches.

Chapter 2 describes the background of this thesis. This chapter begins with a general introduction of service composition and quality of service (QoS), and presents previous work on parallel execution of web services to improve the services' processing efficiency, one of QoSs. In order to create an overview on parallel execution in web service environments, this chapter classifies the existing work into three groups according to the parallelization techniques used in Workflow Management Systems (WfMSs): data parallelism, task parallelism, and pipeline parallelism.

Chapter 3 depicts a way to model parallel execution policies of atomic services. In an environment where service providers employ policies that arbitrarily limit parallel execution of their services, service users' excess parallel execution of the services decreases the processing efficiency of their whole set of tasks. Therefore, the service users need to know the optimal degree of parallelism in order to maximize the processing efficiency of the tasks before invoking the services. To this end, by measuring the effects of the degree of parallelism on processing efficiency of more than 50 services (hereafter *atomic services*), this chapter classifies the parallel execution policies into three types: *low acceleration policy* that gradually decreases the degree of speedup, *steady policy* that fixes the processing efficiency when the degree of parallelism is beyond a certain amount, and *penalty policy* that decreases the processing efficiency as the degree of parallelism increases. Moreover, this chapter also details a way to capture parallel processing efficiency of atomic services that employ a combination of different policies. A series of experiments on 50 different atomic services shows that the proposed model can estimate the parallel processing efficiency with a lower standard error than the existing curve fitting with linear and quartic regression.

Chapter 4 presents a way to predict the parallel processing efficiency of composite services. A composite service is a service where a workflow

orchestrates atomic services with different policies. In order to maximize the parallel processing efficiency of composite services, service users need to optimize the degree of parallelism by considering policies of all atomic services. Therefore, this chapter introduces data parallelism and pipeline parallelism to execute a workflow in parallel. Processing time-line of the pipeline parallelism is used to define an aggregation function to compute parallel processing efficiency for each simple workflow consisting of single control construct, such as a sequential construct, concurrent construct, conditional construct, and loop construct. This chapter also proposes a method to synthesize the policies of atomic services in a complex workflow consisting of an arbitrary combination of control constructs. Finally, this chapter evaluates accuracy of the proposed method in predicting the optimal degree of parallelism of composite services. A series of experiments on composite services combining several different translation services shows that the proposed method has good prediction accuracy in identifying optimal degrees of parallelism for composite services.

Chapter 5 proposes a service platform architecture to control parallel execution of composite services based on parallel execution policies of atomic services. Using the proposed method of predicting optimal degree of parallelism of composite services, this chapter designs architecture for controlling parallel execution of composite services. The architecture first analyzes parallel execution policies of atomic services that compose the composite service. It then computes the optimal degree of parallelism of the composite service. Finally, the architecture generates a parallel execution configuration file that is interpreted by an extended workflow engine to control parallel execution of the composite service. Furthermore, in order to efficiently process multiple workflows that share the same atomic service, the architecture re-calculates optimal degree of parallelism of each workflow by considering multiple requests from different workflows sent to the shared atomic service. It then updates the parallel execution configuration files with the new optimal degree of parallelism for the workflows at run-time. To verify the effect of the architecture in maintaining optimal parallel processing efficiency of workflows, the architecture is implemented in the Language Grid, a service platform that is specialized in natural language processing. An experiment is conducted on multiple language composite services; the results show that the proposed architecture can significantly improve the parallel processing efficiency of composite services.

Chapter 6 concludes the thesis by summarizing original contributions toward a policy-aware parallel execution control system for enhancing parallel processing efficiency of composite services and also suggesting possible future directions.

注）論文内容の要旨と論文審査の結果の要旨は１頁を３８字×３６行で作成し、合わせて、３，０００字を標準とすること。
論文内容の要旨を英語で記入する場合は、４００～１，１００wordsで作成し審査結果の要旨は日本語５００～２，０００字程度で作成すること。

（論文審査の結果の要旨）

本論文は、並列実行によるWebサービスの高速化を目的とするものである。そのために、まずサービス提供者の並列実行に関するポリシーをモデル化し、最適な並列度となるよう、サービス利用者がWebサービスを実行する手法を提案している。得られた主要な成果は以下の通りである。

**1. 原子サービスの並列実行ポリシーのモデル化**

サービス提供者のポリシーによってサービスの並列実行が人為的に制限される環境では、利用者による過剰なWebサービスの並列実行は、むしろタスク全体の処理効率を低下させる。そこで、サービス実行に先立ち、タスク全体の処理効率を最大化する並列度を知る必要がある。そのために、予め約50種類の実サービス（原子サービスと呼ぶ）の処理効率と並列度の関係を測定し、並列度を上げると処理効率向上の度合いが徐々に低下する「減加速ポリシー」、所定の並列度以上の処理効率を一定とする「定常ポリシー」、並列度を上げるとむしろ処理効率が悪化する「処罰ポリシー」の三種類に分類している。また、それらの組み合わせで任意のポリシーがモデル化できることを示している。実際に約50種類の原子サービスを対象に評価を行い、提案モデルが線形回帰モデルおよび曲線回帰モデルと比較して、正確に並列処理の効果を推定できることを確認している。

**2. 複合サービスの並列実行効率の予測**

複合サービスは、異なるポリシーを持つ原子サービスを、ワークフローを用いて組み合わせたものである。このため、複合サービスを並列実行により高速化するためには、全ての原子サービスの並列実行ポリシーを考慮して、最適な並列度を決定する必要がある。そこで、ワークフローの制御構造である逐次構造、並行構造、選択構造、反復構造ごとに複合サービスの処理効率を計算する集約関数を定義し、制御構造を任意に組み合わせたワークフローのサービスの処理効率を予測する手法を考案している。実際に複数の翻訳サービスを組み合わせた複合サービスを用いて評価を行った。その結果、複合サービスの処理効率を最大化するために、提案手法が構成要素である各々の原子サービスの並列度を高い精度で決定できることを確認している。

**3. 複合サービスの並列実行制御アーキテクチャの実装**

提案した複合サービスの処理効率の予測手法を用い、複合サービスの並列度を制御するサービスプラットフォームのアーキテクチャを設計している。サービスプラットフォームは、複合サービスを構成する原子サービスの最適な並列度を予測手法に基づいて計算し、並列実行を制御する。また、複数の利用者によるサービス実行を効率よく処理するために、同一の原子サービスを共有する複合サービス間で最適な並列度を割り当てる。実際に，提案したアーキテクチャを自然言語処理に特化したサービスプラットフォームである言語グリッドに実装し、その効果を検証している。

以上、本論文は、原子サービスの並列実行ポリシーのモデル化、複合サービスの並列実行効率の予測、複合サービスの並列実行制御アーキテクチャを提案し、並列実行によるWebサービスの高速化に寄与している。よって、本論文は博士（情報学）の学位論文として価値あるものと認める。また、平成28年2月24日に実施した論文内容とそれに関連した試問の結果、合格と認めた。

注）論文審査の結果の要旨の結句には、学位論文の審査についての認定を明記すること。更に、試問の結果の要旨（例えば「平成　年　月　日論文内容とそれに関連した口頭試問を行った結果合格と認めた。」）を付け加えること。

Webでの即日公開を希望しない場合は、以下に公開可能とする日付を記入すること。
要旨公開可能日：　　年　　月　　日以降